

Proposing an Algorithm to Solve the Forward and Reverse Logistics Distribution Problem with One Door Container

Stephen Sanjaya Budi¹, Paulina Kus Ariningsih^{1*}, Titi Iswari¹

Abstract: Forward and reverse logistics are two types of distribution methods that shall be synergized in practices. Two problems in synergizing the two distributions type are (1) how to route vehicles and (2) how to pack the goods inside the vehicle. A truck with only one door for loading and unloading process could create numerous problems of item packing activities. An item picked up from a customer could occasionally block other goods which need to be delivered; hence, the courier shall unload other items before the loading process. This condition will increase the probability of item damage, longer on-loading/off-loading (lo/lo) time, and higher lo/lo cost because of the rapid item movement. Therefore, this article aims to propose an algorithm to solve the problem by creating an algorithm hybrid of routing and packing to find the solution for routing and packing problem, sequentially, with a metaheuristic approach. The proposed method calculates the cost from routing procedure and sum of item movement in every loading and unloading process. Based on the trial on 25 cases, this algorithm generates 59.64% of the containers have zero goods repacking. Several potential future research avenues are also proposed in this article.

Keywords: Supply chain; reverse logistics; loading problem; genetic algorithm; VRP-SPD.

Introduction

The supply chain can be defined as an interconnecting group of companies which can increase the value in the process of changing the input until the end-product based on customer demand (Lu [1]). This system includes logistics, both forward and reverse logistics. Forward logistics is an activity where the supplier will deliver raw materials or product to their customer, while the antithesis process from forward logistics is called reverse logistics (Simchi-Levi *et al.* [2]). Govindad and Soleimani [3] stated that when those activities are done simultaneously; forward and reverse logistics system can perform as a closed-loop supply chains system. In routing problem, simultaneous forward and reverse logistics is named VRP-SPD (Vehicle Routing Problem with Simultaneous Pickup and Delivery). Pick-up activity in routing problem refers to an activity of receiving goods from point of pick up to be sent to vehicle's depo which is equal to reverse logistics. Montane and Galvao [4] states that the problem of forward and reverse logistics can be solved by VRP-SPD. Routing and packing are two problems in logistics operations.

Adventia [5] found that loaded goods are often moved during unloading and loading processes, which increase the probability of damage. The complexity of the problem arises when the truck used in the delivery process has one access (door) to move the item in and out the vehicle. A vehicle has a specific route to serve. In many times, a vehicle would have to deliver and pick up some reversed goods. On occasion a picked-up item from a customer blocks other goods that need to be delivered; hence, the courier shall unload other items and therefore increasing the quantity of movement.

Dessaulniers *et al.* [6] proposed a mathematical model to solve the Vehicle Routing Problem with Pick-up and Delivery (VRP-PD) that describes how to route vehicle for multi delivery and pick-up spots. Fan [7] developed the model to emphasize the importance of waiting time toward customer satisfaction by proposing VRP-PD with time windows (VRP-PDTW). Hosny and Mumford [8] showed the solution to this problem is by minimizing the distance subjected to loading time. Meanwhile, Grandinetti *et al.* [9] proposed a mathematical model with a multi-objective approach: Minimizing distance, travel time, and the number of vehicle.

The main problem with packing is how to maximize vehicle capacity. Pedrucci *et al.* [10] showed three types of packing problem, i.e., 3D-BPP (three dimensional-bin packing problem), 3D-CLP (three dimensional-container loading problem), and 3D-CVRP (three dimensional-container vehicle routing problem). The 3D-CVRP model uses routing problem for the first input.

¹ Faculty of Industrial Technology, Department of Industrial Engineering, Universitas Katolik Parahyangan, Jl. Ciumbuleuit 94, Bandung, Indonesia 40141

Email: sanjayabudi.stephen@gmail.com; paulina.ariningsih@unpar.ac.id; titiiswari@unpar.ac.id

* Corresponding author

Many studies, as have been described above, have proposed mathematical models or algorithms to solve the problem separately, for routing or packing. However, the problem needs to be solved to reduce the frequency of items' movement during lo/lo. Therefore, a sequential algorithm is designed to find a solution.

Martello [11] had proposed a tabu-search algorithm to solve the Vehicle Routing Problem with Container Loading Problem (VRP-CLD) simultaneously. However, the condition is not applicable for pick-up and delivery. Bortfeldt and Homberger [12] proposed an algorithm to be used in pick-up and delivery. In the algorithm, item position for every customer is placed first, and then the best route for the vehicle is solved by minimization of the distance. The last process is combining the best route and packing position for every customer in the specific route. However, sometimes the algorithm is unable to be used in one door container.

The method was further developed by Ariningsih [13] which emphasized the ease of practical implementation, resulting in the routing processed firstly then continued with the packing process would be possible to solve the simultaneous loading and routing problem. However, the algorithm was made only for a specific study case, which was plastic furniture. By adopting the No-Free-Lunch theorem for optimization that mentioned that an algorithm which is applied to specific system might not dominates to all system (Domingos [14]) Thus, there is a need to develop more general algorithm which can be used for various type of goods, especially for goods which is delivered by one accessed container. In the meantime, there is lack of reference that is suitable to solve the literature gap especially for vehicle with single accessed container. Thus, this article intends to explore an algorithm to solve the routing and packing problem simultaneously especially for one accessed container. This exploration would use a single type of product packaging and weight.

In this study, the improvement to solve of proposed algorithm is also developed with metaheuristic approach: Genetic Algorithm (GA). GA is powerful metaheuristic that was already implemented on both container loading problem and the VRP itself (Domingos [15], Erdem [16], Gehring, and Bortfeldt [17]). Therefore, this article is also would like to build proposed algorithm using GA.

This article is structured into six major sections. The first section describes the introduction which informs the problem backgrounds and previous research. The second section will describe the literature studied as background of this research. The third

section is research methodologies. The fourth section narrates the development of the algorithm. The fifth section will explain the implementation of the algorithm, the improvement on solving the algorithm, and research limitation. The last section will summarize the study and propose future potential research.

Methods

In this section, we present study of literature used in this article covering: VRP, CLP, and Genetic Algorithm.

Vehicle Routing Problem (VRP)

VRP is introduced by Dantsig and Ratzer [18] who was showing that VRP is part of NP-hard problem which aims to minimize transportation route. As most discussed combinatorial optimization problem in logistics and transportation area, VRP calculation has developed to accommodate the constraints appeared on real world problem (Pollaris *et al.* [19]). Capacitated Vehicle Routing Problem (CVRP) is one of the most well-known type of VRP [17]. CVRP put a capacity constraint for each truck used for delivery. The extension of CVRP is VRP-SPD where is also commonly find in package delivery transportation services. Rieck and Zimmermann [13] had developed mathematical model to solve VRP-SPD using MILP stated below:

For a VRP-SPD is defined in computer graph $G(V, A)$ with $V = \text{set of node}\{0, 1, 2, \dots, n\} = C \cup \{0\}$ is a set node, and $A = \{(i, j) | i, j \in V\}$ is the arc set. Then $i \in C$ represent customers, while node 0 represents depot. The c_{ij} is associated with arc $\langle i, j \rangle \in A$. A set of K identical vehicle with capacity Q is available at the depot. Each customer $i \in C$ is associated with delivered demand d_i and picked up demand p_i where $d_i + p_i \leq Q$ and for $C = 0$ we set: $d_0 := p_0 := 0$.

The decision variable of the VRP-SPD is:

$$x_{ij} = \begin{cases} 1, & \text{if arc } \langle i, j \rangle \text{ is part of the solution} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

with $i, j \in V$

Meanwhile, the formulation of VRP-SPD:

$$\text{Minimize } z = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2)$$

subject to:

$$\sum_{i \in V} x_{ij} = 1, j \in C, i \neq j \quad (3)$$

$$\sum_{i \in V} x_{ji} = 1, j \in C, i \neq j \quad (4)$$

$$\sum_{i \in C} X_{0i} \leq K \quad (5)$$

$$ld_i \geq ld_j + d_i - M_1(1 - x_{ij}), i \in V, j \in C \quad (6)$$

$$lj \geq ld_j - d_j + p_j, i \in V, j \in C \quad (7)$$

$$lj \geq lj - d_j + p_j - M_2(1 - x_{ij}), i, j \in C \quad (8)$$

$$d_i \leq ld_i \leq Q, i \in V \quad (9)$$

$$p_j \leq l_j \leq Q, j \in C \quad (10)$$

$$x_{ij} \in \{0,1\}, i, j \in V \quad (11)$$

with,

c_{ij} = cost transport which depends on distance from nodes i to j

d_i = demand that should be delivered to node i

p_i = demand that should be pick-up from node i

l_i = amount of shipment load after visiting node i

ld_i = amount of shipment load that must be delivered to node i and all another following node.

The eq. (2) refers to objective function of which is to minimize the total transportation costs which can also means to minimize total distance travelled inside a route. Each consumer shall only visit once by a vehicle as reflected by constraints (3) and (4). The k number of route generated is restricted by Constraint (5). Meanwhile, constraint (6) ensures that a vehicle must not travel in sub-tour, several delivery amounts shall be loaded at the depot. To specify the amount of shipment for a vehicle after the visit of customer i to other customer in the route is referred by inequalities (7) and (8). The large multiplier (big M) is used to make constraint linear as used in (6) and (8). Big M is disjunctive constraint. Constraint (9) and (10) are stated to ensure the capacity constraint is fulfilled. It means that the total amount of demand loaded by a vehicle will not exceed its capacity. Constraint (11) is a binary constraint for decision variables.

Container Loading Problem (CLP)

Another problem in packing is how to maximize the capacity of transporter. Pedruzzi *et al.* [10] wrote three types of problem which are 3D-CLP (*three dimensional-container loading problem*), 3D-BPP (*three dimensional-bin packing problem*), and 3D-CVRP (*three dimensional-container vehicle routing problem*) [10]. The approach of 3D-CVRP is using *routing* as the first starter [10]. The solution of this problem is processed by heuristics model by Bortfeld dan Homberger by finding the packing upon the routing (Bortfeldt and Homberger [12]). The 3D-BPP is aiming to find the maximizing the capacity of a truckload. Single bin-size bin packing problem (SBSBPP) is a method to solve packing for a similar vehicle capacity which can be used for heterogeneous goods (Zhao *et al.* [20]). As written by Pedruzzi *et al.* [10], the basic goal of the models is: (1) Fulfilling volumetric capacity on a truckload. (2) The arrangement of goods is not overlapping over one goods to another. (3) Goods position shall be orthogonal with the vehicle's axis. (4) Delivery activities are following LIFO. (5) Truck load stability.

Table 1. Packing problem terminology by Wäscher *et al.* [21]

Objective functions	Size of transported goods		
	Identical	Slightly different	Completely different
Maximizing output value of transported goods inside the vehicle	Identical item packing problem	Placement problem	Knapsack problem
Minimizing input value of transported goods inside the vehicle	Open dimension problem	Cutting stock problem	Bin packing problem

Meanwhile, Wäscher *et al.* [21] had written different type of terminology as seen on Table 1.

The different terminology is affected by objective function and by the size of the goods. In this study, the packing problem terminology is used to describe the arrangement of goods inside the truckload for loading on and loading off during delivery or pick up activities.

Genetic Algorithm (GA)

Norvic stated that since 1950's, researchers had used *artificial intelligence* (AI) for simulating the real-world case (Norvic [22]). One of the AI methods is *genetic algorithm* (GA) which is found by John Holland in the 1960 (Kramer [23]). John Holland simulated Darwinian 'survival of the fittest' and crossover, recombination, mutation, and inversion for solving mathematical problem. Therefore, GA is called as *population-based algorithm*.

There are four main processes of GA (Kramer [23]) which are selection, *crossover*, *mutation*, and *sampling*. Selection is a process to choose the best individual based on its fitness value. Crossover is a method for finding solutions by combining two individuals. Meanwhile, mutation is evolution of individual which is done by changing the gene (chromosome) structure of each individual. Sampling is a process of creating new generation as successor of previous generation. The pseudocode of GA can be seen in Figure 1 (Kramer [23]).

Kramer [23] stated that elitist selection operators is one among several ways to apply selection process in GA. In mating method, several numbers of good individual would be chosen to replace individual with poor fitness value. Meanwhile, the other individual which are not chosen will still be processed for the next generation without selection process.

Crossover and mutation process are undertaken for finding better solution than previous generation. Before executing the crossover and mutation, encoding process shall be performed. Encoding process is an activity of representing a solution to be able to be processed by GA algorithm [23].

```

1: initialize population
2: repeat
3:   repeat
4:     crossover
5:     mutation
6:     phenotype mapping
7:     fitness computation
8:   until population complete
9:   selection of parental population
10: until termination condition

```

Figure 1. Pseudo Code of GA (Kramer [23])

Encoding is presenting the solution into the form of chromosome. There are many type of encoding process such as: binary encoding, real code encoding, permutation encoding. The permutation encoding, or order encoding would generate chromosome in the form as route or sequences [23]. The process of crossover and mutation in GA is allowing the disappearance of best solution and individual in a population. Therefore, to anticipate it, elitism method is applied. Elitism method is a process to choose several best solutions. The solutions would be kept away to avoid GA process. The solutions that are kept away would be compared to the best solution generated by a GA iteration.

Research Methods

In this study, proposed algorithm is approached by breaking the problem of sequential packing-routing into two separates sub problem, i.e., routing and packing problem. The routing algorithm will be created by solving a VRP-SPD. This is to make sure the feasibility that a vehicle would be loaded several goods which optimize the vehicle capacity. This approach is also suggested by Hosny and Mumford [8] who stated that the initial solution for multi-vehicle VRP-SPD with time windows could be generated by utilizing routing to create individual vehicle routes. Solutions of routing algorithm will be solved via branch-and-cut method from Rieck and Zimmermann [24]. A total of minimum travel distance and the route taken for every vehicle will be obtained.

Packing algorithm is created by forming a three-dimensional loading problem. The constraint in the packing algorithm is the solution from the routing problem and the vehicle capacity. Furthermore, the packing problem is answered by an algorithm capable of finding the position for every item in the container and the movement frequency in every loading and unloading process. The algorithm will be enhanced with genetic algorithm (GA) in finding the solution. The improvement is done to help the optimization process since the designed packing algorithm can only find a single solution. Its eminence and its flexibility to solve many types of problem contribute to the decision to choose GA.

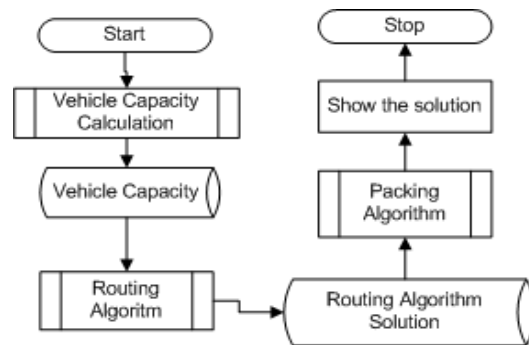


Figure 2. Main flowchart of the proposed algorithm

Based on Bajpai and Kumar's [25] explanation, the general construction and its agility make GA as one of the best algorithms. GA has some parameter which can affect the performance which is the population number, iteration number, crossover, mutation, and selection. All parameter value will be tested to find the best parameter combination which results in the best algorithm performance.

Algorithm implementation is done to identify if the algorithm can solve the forward and reverse logistics with one door (access) constraint. It also shows the way parameters affect the algorithm and its performance. The algorithm is implemented in 25 set data as shown in Rick and Zimmermann [24]. The analysis is carried out not only for every step in the algorithm design process, but also for the result. The parameter to create the proposed algorithm is analyzed as well.

Development of Algorithm

The proposed algorithm is designed to solve the problem on forward and reverse logistics system. Figure 2 illustrates the flowchart of the main algorithm. On the main algorithm, as shown in Figure 3, it is shown that the routing problem is solved first.

The solution of the routing problem is the route for every vehicle, which later used as input for the packing algorithm. A mathematical model of VRP-SPD and the branch-and-cut method designed by Rieck and Zimmermann [24] with CPLEX Studio IDE 12.8.0 are used to unravel the routing problem. Figure 4 shows the flowchart of the routing algorithm. This algorithm is following [23] therefore when the implementation of the algorithm shows same result as [23], the algorithm is validated and verified.

An algorithm to find the solution of movement frequency and item position in the vehicle for every unloading and loading process is used to answer the packing problem. The process of branch and cut can be seen in [26]. Figure 4. illustrates the flowchart of the packing algorithm.

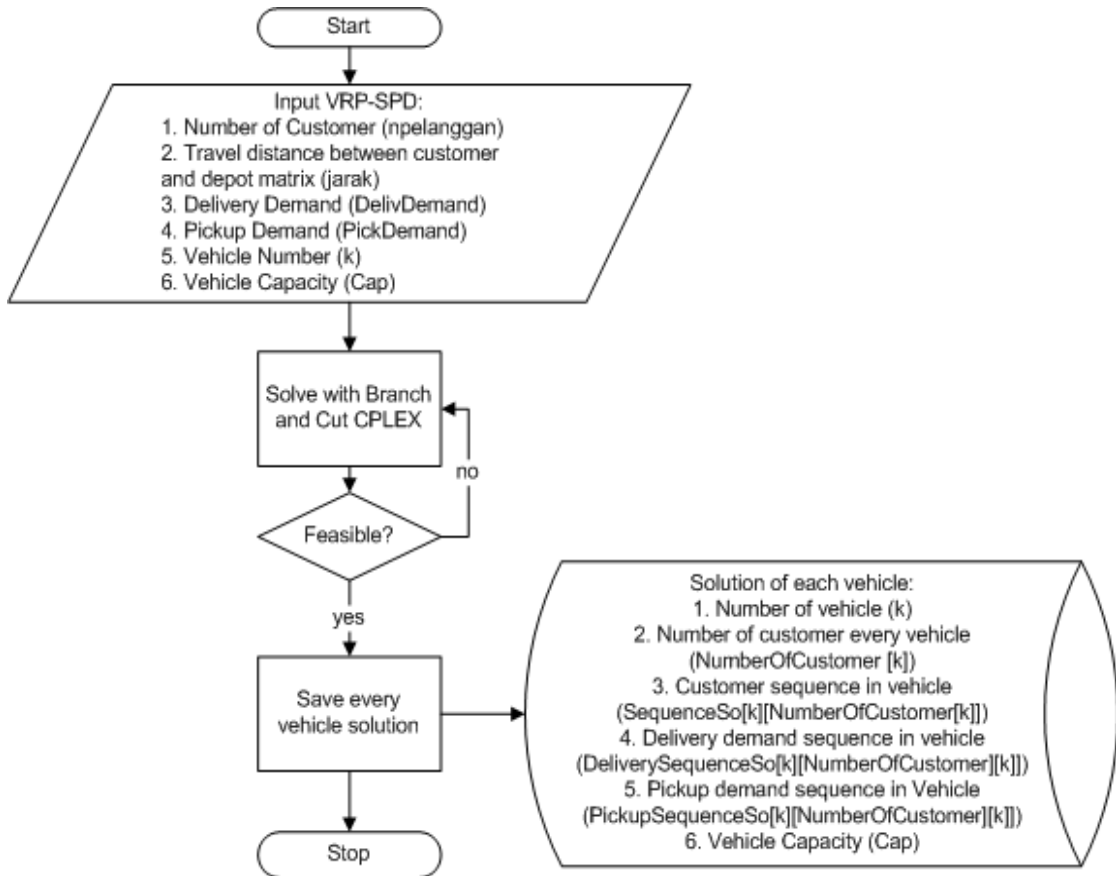


Figure 3. Routing algorithm flowchart

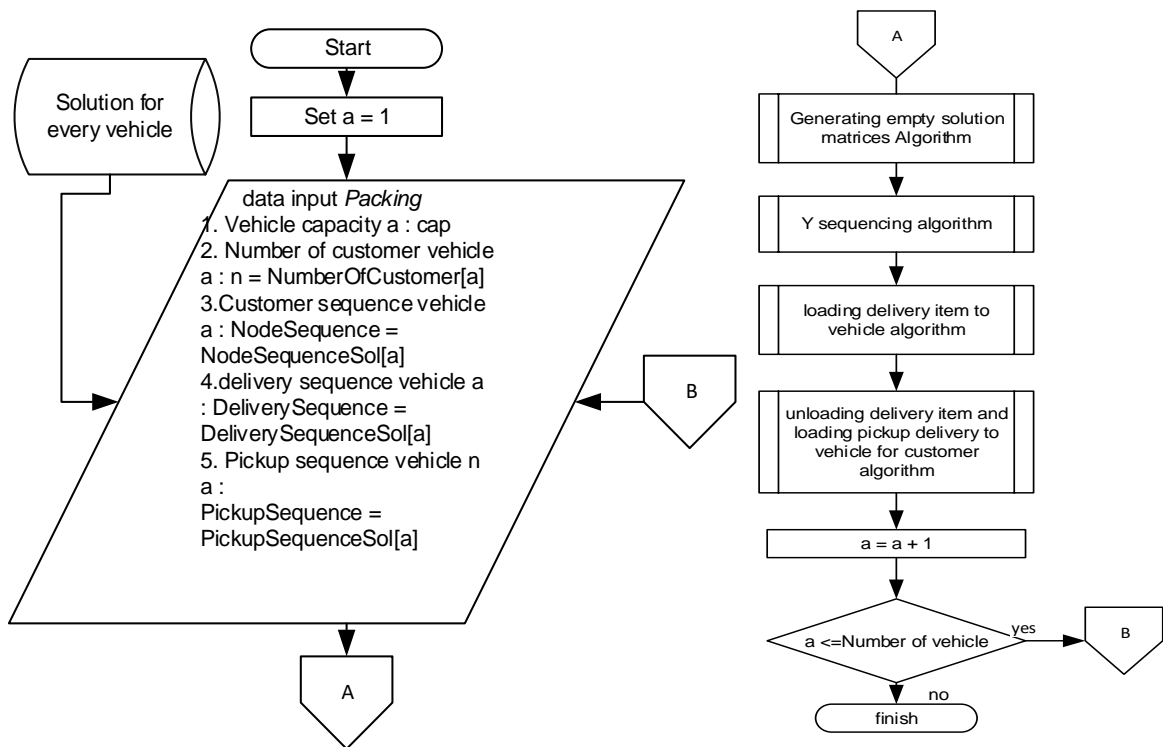


Figure 4. Packing algorithm flowchart

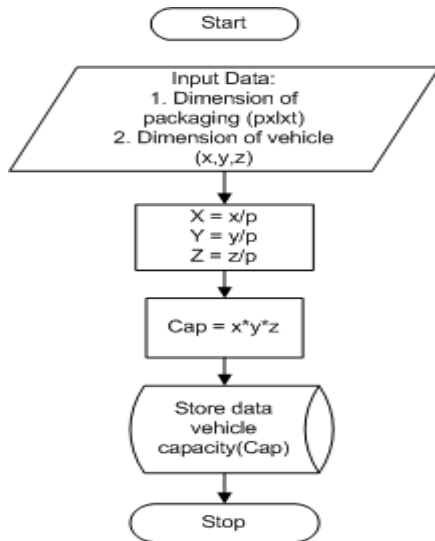


Figure 5. Calculation process of vehicle capacity

Initially, the packing algorithm is started by creating an empty solution matrix. In the process, the matrix size will be determined by vehicle capacity data. Vehicle capacity is obtained based on the dimension of packaging and dimension of vehicle as per seen in Figure 5. This process shall be determined as first step of proposed algorithm.

Supposed that the packaging dimension is $1 \times 1 \times 1$, and dimension of vehicle is $5 \times 4 \times 4$, then the X-length will be five items, Y-length is four items, and Z-length is four items, and vehicle capacity is 80.

Sorting y will be the next step. This process is needed because the loading process will be done from $Y=1$ to $Y=$ the last y-number. Furthermore, it is continued to $y = two$ to $y =$ to $Y-1$. Figure 6 reveals the illustration of packing inside a container during lo/lo.

In logic A, unloading process is done to $Y=3$. After the item is unloaded, there are four more items to be delivered to the next customer. If the picked-up items in the current customer are less than the delivery items to the next customer, then the process is done, and no movement is necessary. However, there will be two movements required to deliver the item to the next customer. In logic B, the quantity of pickup item is more than the number of the subsequent delivery items. In that case, the delivery items will be unloaded first, later followed by the loading of pickup items. After the picked-up items are loaded, the delivery item for the next customer can be reloaded.

Whenever the packing algorithm has produced a result that is not violating the limitations set in the logics, it means that the algorithm and program are verified and validated.

Both routing and packing designed algorithm are simulated by software. The routing algorithm is simulated by using CPLEX programming language, and packing algorithm is simulated using JAVA programming language and Netbeans IDE 8.2.

Results and Discussions

This section will describe the implementation of algorithms for the particular dataset. The algorithm and simulation program has been verified and validated. The verification and validation are done to understand if the algorithm has followed the right logic and the algorithm can solve the problem. Through the implementation, we can see that all algorithms had not broken the limitations. Therefore, the whole algorithm is verified and validated.

Implementation of Routing Algorithm

Routing algorithm implementation is done by using a dataset from Rieck and Zimmermann [24]. There are five data sets, i.e., R121, R141, R161, R181, and R1101. Every set of data have combination with a number of customer and capacity of the vehicle. There are three numerical types of customers, i.e., 15, 17, 20; and two types of vehicle capacity, i.e., 18 and 120. Table 2 is the result of the minimum travel distance and number of route per data set. The routing result stated in Table 2 has been compared to the solution from Rieck and Zimmermann [24] and proved that each solution is the optimum value for every problem. Other than travel distance, the route solution for each vehicle is also obtained and can be seen in Table 3. Presentations of both tables are to give an illustration on the result of routing algorithm which will be used on next step, packing algorithm.

Packing Algorithm Implementation

Packing algorithm implementation is done by using the route that has been obtained from the routing algorithm. Based on the packing algorithm, movement frequency has also been acquired for every vehicle. The recapitulation for movement frequency is exhibited in Table 4. As seen in Table 4, the average movement for packing algorithm is 2.73.

Average movement frequency of Table 4 is 2.7368 movements. The average computation time for running each problem set is 0.000378-seconds.

The position of goods and their movement in every customer is required to make lo/lo planning. Figure 7 shows an example of the packing position. The X-rows show the item position from the back of the container to the front.

The Y-columns show the item position from left to right, and the Z-columns show the item position from bottom to top. In the figure, it is seen that goods for delivery (white boxes, coded D) shall not being blocked by picked-up goods (black boxes, coded P). For example, in customer 2, x3, y3, first and second

layer (z1, z2) from the bottom are filled with picked up goods: P2, but the third and fourth layer (z3, z4) are filled with delivered goods: D1. Thus, when vehicle reach customer 1, D1 is taken out easily, and x3, y3, z3 and x3, y3, z4 can be filled with picked up goods from customer one: P1 easily.

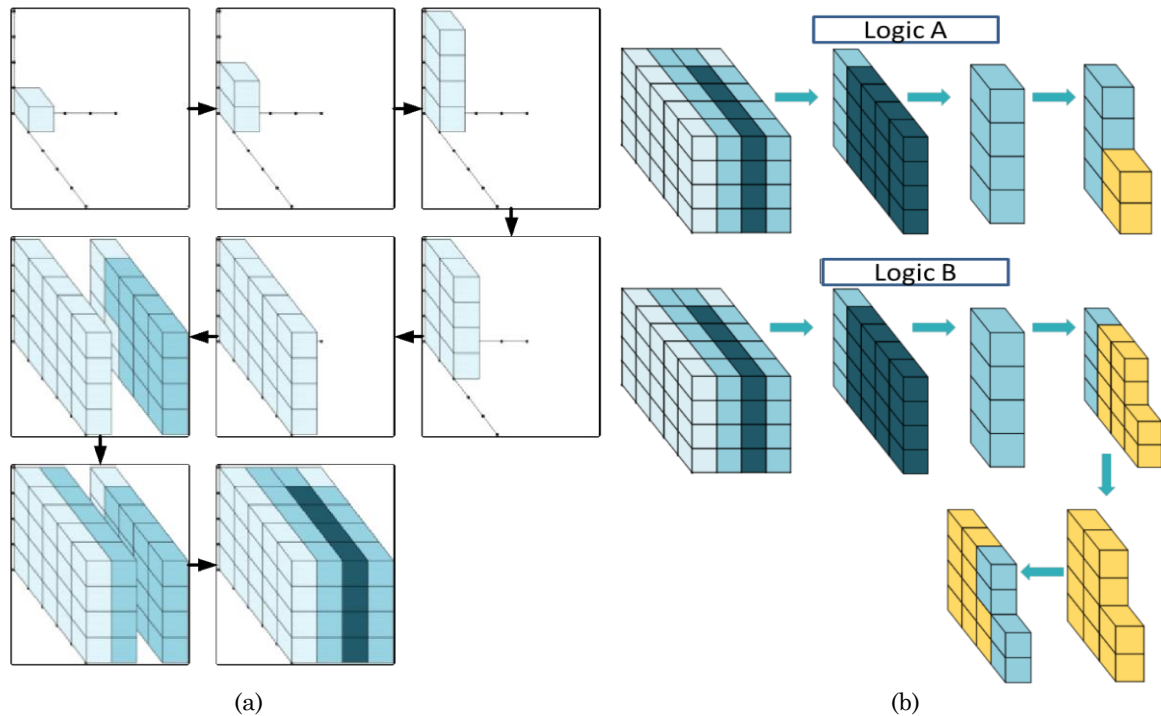


Figure 6. (a) Item loading process (b) Item unloading process

Table 2. Routing distance solution

No.	Data set code	Total Cost (distance unit)	Number of route	No.	Data set code	Total Cost (distance unit)	Number of route
1	R121_15_120	542.199	2	15	R161_20_120	1279.508	2
2	R121_15_80	610.8	3	16	R181_15_120	1755.945	2
3	R121_17_120	564.385	2	17	R181_15_80	1968.38	2
4	R121_17_80	726.062	4	18	R181_17_120	1786.749	2
14	R161_17_80	1211.211	3	25	R1101_20_120	2119.536	2

Table 3. Routing route solution

No.	Dataset code	Vehicle	Route	No.	Dataset code	Vehicle	Route		
1	R121_15_120	1	0-1-8-2-13-10-6-12-15-4-0	14	R161_17_80	1	0-13-6-17-15-1-7-10-0		
		2	0-9-7-5-3-11-14-0			2	0-3-12-5-14-8-9-11-2-4-0		
2	R121_15_80	1	0-1-8-2-13-10-6-12-0	15	R161_20_120	3	0-16-0		
		2	0-7-15-4-9-0			1	0-13-6-17-1-15-19-10-7-20-0		
		3	0-14-11-3-5-0			2	0-3-12-5-18-14-8-9-11-2-4-16-0		
3	R121_17_120	1	0-9-4-15-7-5-3-11-14-0	16	R181_15_120	1	0-15-13-5-4-10-1-6-9-14-3-0		
		2	0-1-8-16-2-17-13-10-6-12-0			2	0-7-11-12-8-2-0		
4	R121_17_80	1	0-14-11-3-5-0	17	R181_15_80	1	0-6-1-14-3-7-11-12-2-0		
		2	0-9-7-1-0			2	0-9-10-4-5-13-15-8-0		
		3	0-8-16-2-17-0			18	R181_17_120	1	0-16-6-1-10-4-5-13-15-0
		4	0-4-15-12-6-10-13-0					2	0-9-17-14-3-7-11-12-8-2-0
11	R161_15_120	1	0-13-6-3-12-5-14-8-9-11-2-4-0	25	R1101_20_120	1	0-14-13-10-1-2-9-16-3-18-15-0		
		2	0-10-15-1-7-0			2	0-12-11-7-6-19-5-4-8-17-20-0		
12	R161_15_80	1	0-13-6-15-1-7-10-0						
		2	0-3-12-5-14-8-9-11-2-4-0						

Table 4. The movement frequency solution for packing algorithm

No.	Dataset code	Vehicle	Movement Frequency	No.	Dataset code	Vehicle	Movement Frequency
1	R121_15_120	1	0	14	R161_17_80	1	2
		2	0			2	7
2	R121_15_80	1	0	15	R161_20_120	3	0
		2	3			1	0
		3	0			2	0
3	R121_17_120	1	0	16	R181_15_120	1	6
		2	0			2	0
4	R121_17_80	1	0	17	R181_15_80	1	4
		2	0			2	0
		3	0			1	0
		4	0			2	4
5	R121_20_120	1	2	19	R181_17_80	1	0
		2	9			2	0
6	R141_15_120	1	6	20	R181_20_120	3	0
		2	16			1	0
		1	0			2	4
7	R141_15_80	2	3	21	R1101_15_120	1	0
		3	0			2	14
		1	7			1	9
8	R141_17_120	2	0	22	R1101_15_80	2	0
		1	0			1	7
9	R141_17_80	2	0	23	R1101_17_120	2	0
		3	11			1	6
		1	6			2	3
10	R141_20_120	2	16	24	R1101_17_80	1	0
		1	0			2	0
11	R161_15_120	1	0	25	R1101_20_120	1	0
		2	0			2	0
12	R161_15_80	1	0			1	0
		2	7			2	0
13	R161_17_120	1	0			1	0
		2	4			2	0

Customer	Axis	z1				z2				z3				z4			
		y1	y2	y3	y4	y1	y2	y3	y4	y1	y2	y3	y4	y1	y2	y3	y4
0	x1	D3	D1	D1	D3	D3	D1	D1	D3	D3	D1	D1	D3	D3	D1	D1	D3
	x2	D3	D1	D2	D3	D3	D1	D2	D3	D3	D1	D2	D1	D3	D1	D2	D1
	x3	D3	D1	D2	D1	D3	D1	D2	D1	D3	D1	D2	D1	D3	D1	D2	D1
	x4	D3	D1	D2	D1	D3	D1	D2	D1	D3	D1	D2	D1	D3	D1	D2	D1
	x5	D3	D1	D2	D1	D3	D1	D2	D1	D3	D1	D2	D1	D3	D1	D2	D1
2	x1	D3	D1	P2	D3	D3	D1	P2	D3	D3	D1	P2	D3	D3	D1	P2	D3
	x2	D3	D1	P2	D3	D3	D1	P2	D3	D3	D1	P2	D1	D3	D1	P2	D1
	x3	D3	D1	P2	D1	D3	D1	P2	D1	D3	D1	D1	D1	D3	D1	D1	D1
	x4	D3	D1	D1	D1	D3	D1	D1	D1	D3	D1	0	D1	D3	D1	0	D1
	x5	D3	D1	0	D1	D3	D1	0	D1	D3	D1	0	D1	D3	D1	0	D1
1	x1	D3	P1	P2	D3	D3	P1	P2	D3	D3	P1	P2	D3	D3	P1	P2	D3
	x2	D3	P1	P2	D3	D3	P1	P2	D3	D3	P1	P2	P1	D3	P1	P2	P1
	x3	D3	P1	P2	0	D3	P1	P2	0	D3	P1	P1	0	D3	P1	P1	0
	x4	D3	P1	P1	0	D3	P1	P1	0	D3	P1	P1	0	D3	P1	P1	0
	x5	D3	P1	P1	0	D3	P1	P1	0	D3	P1	P1	0	D3	P1	P1	0
3	x1	P3	P1	P2	P1	P3	P1	P2	P1	P3	P1	P2	P3	P3	P1	P2	P3
	x2	P3	P1	P2	P3	P3	P1	P2	P3	P3	P1	P2	P3	0	P1	P2	P3
	x3	0	P1	P2	P3	0	P1	P2	P3	0	P1	P1	P3	0	P1	P1	P3
	x4	0	P1	P1	P3	0	P1	P1	P3	0	P1	P1	P3	0	P1	P1	P3
	x5	0	P1	P1	P3	0	P1	P1	P3	0	P1	P1	P3	0	P1	P1	P3

Figure 7. Packing position solution example

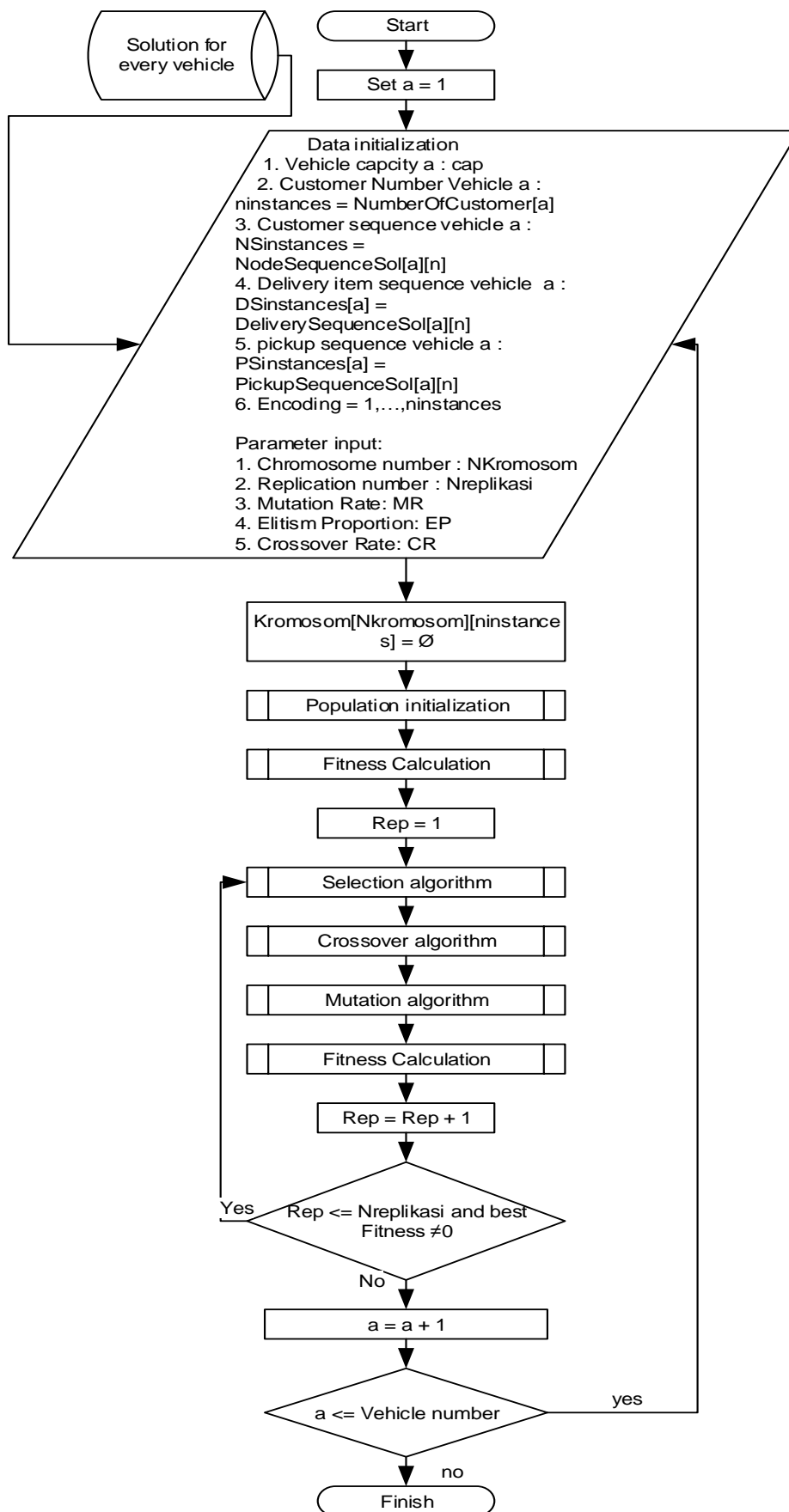


Figure 8. Improvement of the packing algorithm with GA Flowchart

Improvement Solution Procedure of Packing Algorithm using GA

The improvement is done to find a better solution other than a single solution. The packing algorithm is used for determining the fitness value for every chromosome. Figure 8 demonstrates the flowchart of the improvement packing algorithm using GA. The process of GA used in this study is still done with basic step or procedure. The process of GA is as per follow.

The first step in the packing algorithm is initializing the population according to the user input. Secondly, the fitness calculation is started to find the fitness value. The fitness value is the movement frequency. Afterward, the selection process is carried out to determine chromosomes with the best value to be stored and not processed in the next step. Steady state selection method is performed to find the best chromosomes to replace bad chromosomes. The best value chromosome is maintained by elitism proportion.

Crossover process is done to find a new solution by crossing the gen from each chromosome to other. The chromosome which has been crossed over cannot be further crossed over. The method for crossover is uniform crossover which provides the uniformity in combining the bits of both parents (Umbarkar, and Sheth [27]). To avoid duplication of chromosomes, recombination process is performed. The recombination process is finding and replacing the duplicated gene with the missing genes. Number of chromosomes to be crossed over is determined by the crossover rate. The crossover process is not performed for individual which had been maintained through elitism proportion.

Mutation is performed by changing the chromosomes' structure by swap mutation. The number of chromosomes to be mutated is determined by the mutation rate. The next process is the fitness calculation for every chromosome. Selection, crossover and mutations are repeated until the determined number of replication or until the solution with zero fitness value is found.

Encoding and Decoding for Packing Algorithm using GA

Encoding is conducted to translate the packing problem to the GA algorithm, meanwhile decoding is to reverse the translation. The encoding process is performed by symbolizing the chromosome as the route for the delivery sequence.

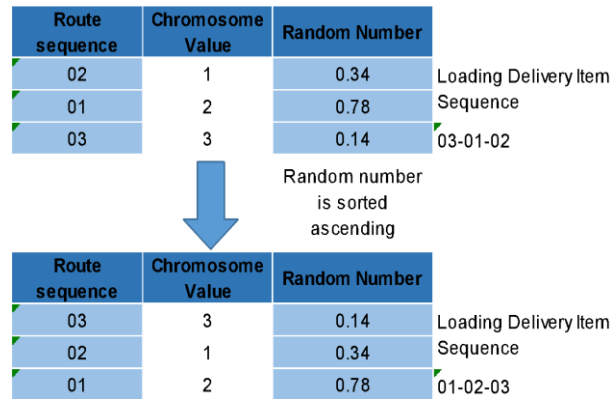


Figure 9. Quality of filter mapping

Table 5. OFAT result

Parameter	Range	Level			
		A	B	C	D
Chromosome number	Integer > 0	10	50	100	500
Replication number	Integer > 0	10	50	100	500
Mutation Rate	real 0 ~ 1	0.25	0.5	0.75	1
Elitism Proportion	real 0 ~ 1	0.25	0.5	0.75	1
Crossover Rate	real 0 ~ 1	0.25	0.5	0.75	1

The route from routing solution will be numbered from 1 to the number of customers served. For every gen in the chromosome will be generated a random number with a uniform distribution (0,1). The random number will be sorted ascending and then the new chromosome will be generated. The encoding and decoding process can be seen in Figure 9.

Parameter Setting

Before the implementation is processed, the value of the parameter needs to be determined. Two steps in determining the parameters are conducted, i.e., one factor at a time method (OFAT) and factorial experiment $2^f \times n$. OFAT determines two of the best level which will be used in a factorial experiment. The factorial experiment later could oversee the effect of the parameter to the algorithm. The response for the experiment is the average of 25 data set value. The measured parameter for OFAT is shown in Table 5.

There are four levels to be tested with OFAT. Each experiment is replicated five times. Steady value for chromosome number and replication number is 10. Steady value for mutation rate, elitism proportion, and replication number is 0.5. Based on OFAT, the best response is marked with grey color in the table. Next is the factorial experiment $2^f \times n$ with two levels, five factors, and five replications. The obtained respond then will be tested with ANOVA on Minitab 18 Application. Based on ANOVA, the treatment which affects the performance is shown in Table 6.

Table 6. Affecting parameter

Parameter	P-Value
JKromosom	0
MR	0
EP	0
CR	0
JKromosom*MR	0
JKromosom*EP	0
JKromosom*CR	0
MR*EP	0.001
EP*CR	0
JKromosom*JReplikasi*CR	0.024
JKromosom*MR*EP	0.001
JKromosom*EP*CR	0.001
JReplikasi*MR*CR	0.016
JKromosom*JReplikasi*MR*CR	0.046

Table 7. Parameter value

Parameter	Level
Chromosome number	500
Replication number	100
Mutation Rate	1
Elitism Proportion	0.25
Crossover Rate	1

The longest computational time is 839.10 second. To determine the best level, an analysis to interaction plot needs to be conducted. The highest level of interaction plot is an interaction between chromosome number, replication number, mutation rate, and crossover rate. Figure 10 shows the four-way interaction plots for the four parameters.

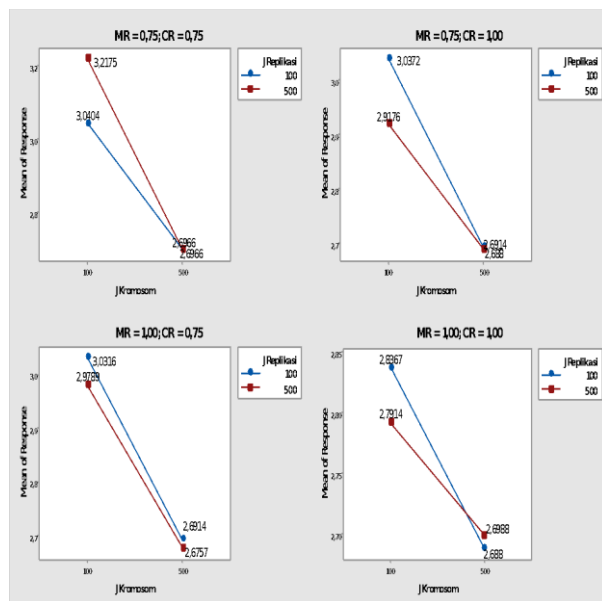


Figure 10. Interaction Plot between chromosome number, replication number, mutation rate, and crossover rate.

According to the interaction plot, there are no parallel lines. It also shows four parameters have no interactions. The best level is chosen by showing the minimum respond which is 2.68. Elitism proportion value will be determined based on the three-way interaction plot between some chromosomes, mutation rate, and elitism proportion.

Figure 11 shows the three-way interaction plot. It can be concluded that the value of 0.25 from elitism proportion can give the best respond than the value of 0.75. The parameter value that will be used in the implementation (see Table 7).

As in Table 7, the best parameter value to obtain the minimum solution are 500 for chromosomes number, 100 for iterations number, 1 for mutation rate, 0.25 for elitism proportion and 1 for crossover rate.

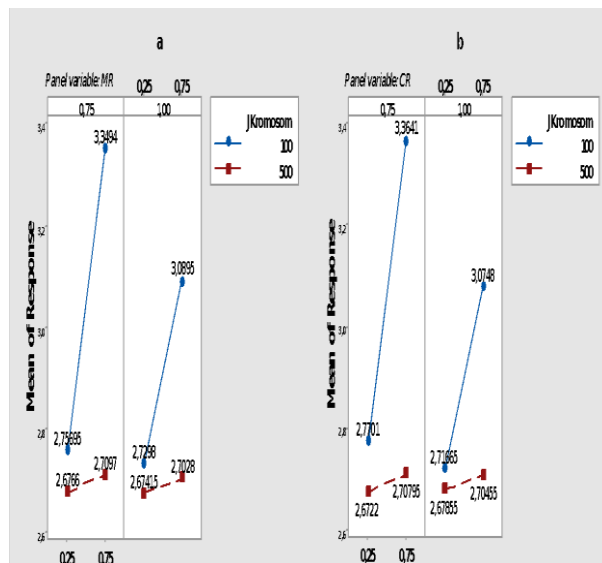


Figure 11. (a.) interaction plot between chromosome numbers, elitism proportion, mutation rate (b.) interaction plot between chromosome numbers, elitism proportion, crossover rate

Implementation of GA

The next step is the implementation of the modified packing algorithm with GA using the obtained parameters. Table 8 shows the result and the comparison with the single solution from packing algorithm using the parameter value stated in Table 7.

Column A shows the single solution of packing algorithm and the column B shows the solution of modified packing algorithm with GA. The unit for column A and B is the total item movement of the vehicle. As shown in Table 8, there are two improvement solutions written in column B, which are found to be better (lower cost of movement) compared to the solution in column A. The improvement solution appears in the sixth and tenth data sets which each data set consist of two vehicles. For each vehicle in average total improvement GA is 6.25%. But in the average, the GA improves 2.2% compared to the ordinary algorithm. This result is, even though appears to be little, showing that the usage of meta-heuristic method, especially GA with the experimented parameters, may improve algorithm performance.

Table 8. Movement frequency solution for implementation of the packing algorithm with GA and the comparison

No.	Data set code	Vehicle	Movement Frequency		Differ-ence	No.	Data set code	Vehicle	Movement Frequency		Differ-ence
			Without GA (A)	With GA (B)					Without GA (A)	With GA (B)	
1	R121_15_120	1	0	0	0	15	R161_20_120	1	0	0	0
		2	0	0	0			2	0	0	0
2	R121_15_80	1	0	0	0	16	R181_15_120	1	6	6	0
		2	3	3	0			2	0	0	0
		3	0	0	0			1	4	4	0
3	R121_17_120	1	0	0	0	17	R181_15_80	2	0	0	0
		2	0	0	0			1	0	0	0
		1	0	0	0			2	4	4	0
4	R121_17_80	2	0	0	0	18	R181_17_120	1	0	0	0
		3	0	0	0			2	4	4	0
		4	0	0	0			1	0	0	0
		1	0	0	0			2	0	0	0
5	R121_20_120	1	2	2	0	19	R181_17_80	3	0	0	0
		2	9	9	0			1	0	0	0
6	R141_15_120	1	6	6	0	20	R181_20_120	2	4	4	0
		2	16	14	-0.125			1	0	0	0
7	R141_15_80	1	0	0	0	21	R1101_15_120	2	14	14	0
		2	3	3	0			1	9	9	0
		3	0	0	0			2	0	0	0
8	R141_17_120	1	7	7	0	22	R1101_15_80	1	7	7	0
		2	0	0	0			2	0	0	0
		1	0	0	0			1	6	6	0
9	R141_17_80	2	0	0	0	23	R1101_17_120	2	0	0	0
		3	11	11	0			1	6	6	0
		1	0	0	0			2	3	3	0
10	R141_20_120	1	6	6	0	24	R1101_17_80	1	0	0	0
		2	16	14	-0.125			2	0	0	0
11	R161_15_120	1	0	0	0	25	R1101_20_120	1	0	0	0
		2	0	0	0			2	0	0	0
12	R161_15_80	1	0	0	0			Mean	2.7368	2.6667	
		2	7	7	0			Standard Deviation	4.23222	4.01930	
13	R161_17_120	1	0	0	0			Minimum	0.00	0.00	
		2	4	4	0			Maximum	16.00	14.00	
		1	2	2	0						
14	R161_17_80	2	7	7	0						
		1	2	2	0						
		3	0	0	0						

Table 9. Chi-square test result

Ho:	There is no difference between Without GA and With GA	
Ha:	There is a difference between Without GA and With GA	
Rejection:	Reject if Asymp. Sig. ≥ 0	
Calculation result:	Without GA	With GA
Chi-Square	158.263	137.368
df	9	8
Asymp. Sig.	0.000	0.000

Small difference happens for GA improvement because of several reasons. In this study, only five GA parameters are tested through OFAT. However, there is another parameter that might influence the fitness in GA, such as: number of iteration. The average of computational time is only 2.58 second, so there is a possibility that the result would be better if the number of iteration is higher where it is usually in contrast with the time needed for doing computation. The selections of parameters used for OFAT are also influencing the optimization of parameter chosen and furtherly, affecting the result. For each chosen parameter, only four levels of parameter were

tested in OFAT. There is a possibility that different parameter set would perform better than best parameter found in OFAT.

A Chi-square test is performed to test whether the result of GA is significantly different with the non-improved algorithm. Result of chi-square test with the p-value 0.05 is as seen in Table 9. It is clearly seen that value of significance $< p$ -value 0.05, therefore we fail to reject Ho and there is no significant difference between the algorithm without improvement and with improvement.

Research Limitation

In this study, we only present the implementation of algorithm towards a single object. Meanwhile in practice, the type of items that should be picked up or delivered by a company might be various. This surely would impact to the solution of packing which should considering the dimension of each various goods. As an extension of VRP-SPD, time windows of operations and vehicle in real practice are also being considerate on routing. The solution of routing would be very different. Therefore, the input for packing section would also be different if the objective of packing algorithm has time constraints.

Another assumption used in this study is that vehicles owned by business are having similar capacity of Q , meanwhile in reality, the vehicles used are having various type of capacity. Considering different size of capacity would surely change the mathematical model of VRP-SPD. The solution would also be different.

Branch and cut algorithm can be used to improve the solution compared to branch and bound [28] therefore it's common to be used for solving the MILP (mixed integer linear programming) or any integer programming [29].

The proposed algorithm, as in Figure 2, itself had proven to be able to produce minimum cost in various dataset, thus, we can argue that the algorithm itself has a foundation to be implemented in any demand dataset for single product and single vehicle capacity. As we might know, that this study is using GA with OFAT parameter determination. The GA parameter given in the implementation sections are tested for 25 data sets of [24] under OFAT. Five parameters affect the modified packing algorithm, i.e., number of chromosomes, number of iterations, mutation rate, elitism proportion, and crossover rate. Furthermore, some interactions are also observed, such as the interaction of chromosomes numbers, number of iterations, mutation rate, and crossover rate. The modified packing algorithm with genetic algorithm to solve the problem pertaining to forward and reverse logistics can obtain better solution if compared to the single solution of packing algorithm. The average result for the modified algorithm is 2.67. It is clearly shown that the proposed algorithm can be improved by GA during finding the solution procedure. However, the improvement resulted is not significantly different. This fact is coherent with the NFL theory.

Conclusion

Based on the experiment, the summary of analysis can then be derived. The routing-first packing-second algorithm has been created to solve the

forward and reverse logistics problem of one access truck constraint. The routing process is done by using the exact method of branch and cut, and the packing process is solved by using a packing algorithm that has been modified with genetic algorithms. The algorithm might become alternatives for the assignment of vehicle as well as the arrangement of goods during lo/lo. Business may receive some productivity by implementing the algorithm. To bring the problem on this algorithm closer to reality, some future potential researches are suggested in this article. Those future works are improving the algorithm for multi-sized product, multi-capacity vehicle set, the solution generation for branch and cut algorithm especially for large datasets.

Acknowledgment

Authors would like to thank to LPPM UNPAR for funding this research through Internal Funds: Dosen Muda Scheme.

References

1. Lu, D., *Fundamentals of Supply Chain Management*. Dr. Dawei Lu & Ventus Publishing ApS. 2011.
2. Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E., *Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies 3rd edition*, McGraw-Hill International Edition. 2008.
3. Govindad, K. and Soleimani, H., A Review of Reverse Logistics and Closed-loop Supply Chains: A Journal of Cleaner Production Focus. *Journal of Cleaner Production*, 142, 2017, pp. 371-384.
4. Montane, F.A.T., and Galvao, R.D., Vehicle Routing Problems with Simultaneous Pick-up and Delivery Service, *Operations Research*. 39(1), 2002, pp. 19-33
5. Adventia, I.C.D., *Perancangan Algoritma Downstream Logistics untuk Produk Plastik di PT. Rentang Buana Niaga Makmur*. Skripsi: Universitas Katolik Parahyangan. 2016
6. Desaulniers, G., Desroriers, J., Erdmann, A., Solomon, M.N., Soumis, F., *VRP with Pick Up and Delivery*, Book Chapters, 2002.
7. Fan, J., The Vehicle Routing Problem with Simultaneous Pickup and Delivery Based on Customer Satisfaction. *Procedia Engineering*, 15, 2011, pp. 5284 – 5289.
8. Hosny, M.I, and Mumford, C.L, Constructing Initial Solutions for the Multiple Vehicle Pickup and Delivery Problem with Time Windows, *Journal of King Saud University – Computer and Information Sciences*, 24, 2012, pp. 59-69.
9. Grandinetti, L., Guerriero, F., Pezzella, F., Piscane, O., The Multi-Objective Multi-Vehicle Pickup and Delivery Problem with Time

- Windows, *Procidia-Social Behavioral Sciences*, 111, 2014, pp. 203-212.
10. Pedruzzi, S., Nunes, L. P. A., Rosa, R. A, Arpini, B.P., A Mathematical Model to Optimize the Volumetric Capacity of Trucks Utilized in the Transport of Food Products, *Gest. Prod*, 23, 2016, pp. 350-364.
 11. Martello, S., *Routing Problems with Loading Constrains (with an Introduction to Vehicle Routing*. Workshop on Traffic Optimization. 2015 retrieved from: <https://wwwproxy.iwr.uniheidelberg.de/groups/comopt/conferences/traffopt2015/Slides/Martello.pdf> on 23 January 2018
 12. Bortfeldt, A. and Homberger, J., Packing First, Routing Second—A Heuristic for the Vehicle Routing and Loading Problem. *Computers & Operations Research*, 40, 3, 2013, pp. 873-885.
 13. Ariningsih, P.K., Sandy, I.A., and Adventia, I.C.D., Pengembangan Sistem Operasional Downstream Logistics dengan Modifikasi P1R2. *Jurnal Metris*, 17, 2016, pp.71-80.
 14. Domingos, P., *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, 2015, Basic Books: New York
 15. Erdem, H.A., Solving Container Loading Problem with Genetic Algorithm, *15th IEEE International Symposium on Computational Intelligence and Informatics*, Budapest, Hungary, 2014, 19-21 November, pp. 391-396.
 16. Gehring, H., and Bortfeldt, A., A Genetic Algorithm for Solving the Container Loading Problem, *International Journal of Operational Research*, 4(5/6), 1997, pp. 401-418.
 17. Iswari, T., Budi, S.S., and Ariningsih, P.K, A Metaheuristic Approach for the Vehicle Routing Problem with Simultaneous Pick-Up and Delivery (VRP-SPD), Presented in *the 19th Asia Pacific Industrial Engineering and Management Systems (2018 APIEMS)* Hongkong, 5-8 December 2018.
 18. Dantzig, G. B. and Ramser, J. H., The Truck Dispatching Problem. *Management Science* 6(1), 1959,pp. 80-91.
 19. Pollaris, H., Braekers, K., Caris, A., Janssens, G.K., & Limbourg, S., Vehicle Routing Problems with Loading Constraints: State-of-the-Art and Future Directions, *OR Spectrum*, 37(2),2015, pp. 297-330.
 20. Zhao, X., Bennell, J.A., Bektas, T., and Dowland, K., A Comparative Review of 3D Container Loading Algorithms, *International Transactions in Operational Research*, 23, 2016, pp. 287-320.
 21. Wäscher, G., Haußner, H., and Schumann, H., An Improved Typology of Cutting and Packing Problems, *Eroupean Journal of Operational Research*, 183 (2007), 2006, pp.1109-1130
 22. Norvic, P, Artificial Intelligence: Early Ambitions. 2012, *New Scientist*, 2016 (2889), pp. ii-iii.
 23. Kramer, O., *Genetic Algorithms Essentials*, Springer International Publishing, 2017.
 24. Rieck, J., and Zimmermann, J., Exact Solutions to the Symmetric and Asymmetric Vehicle Routing Problem with Simultaneous Delivery and Pick-Up. *Business Research*, 6, 1, 2013, pp. 77-92.
 25. Bajpai, P. and Kumar, M., Genetic Algorithm – an Approach to Solve Global Optimization Problems. *Indian Journal of Computer Science and Engineering*, 1(3), 2010, pp. 199 – 206.
 26. Bixby, E. R., and IEE, E., Solving a Truck Dispatching Scheduling Problem Using Branch-and-Cut. *Operations Research*, 46(3), 2000, pp. 355-367.
 27. Umbarkar, A.J., and Sheth, P.D., Crossover Operators in Genetic Algorithms: A Review, *ICTACT Journal on Soft Computing*,6(1), 2015, pp. 1083-1092.
 28. Albert, S., *Solving Mixed Integer Linear Program Using Branch and Cut Algorithm*, 1999, Master Thesis, North Carolina State University.
 29. Winston, W.L., *Operations Research: Applications and Algorithms 4th edition*, 2004, Duxbury Press, South Melbourne.